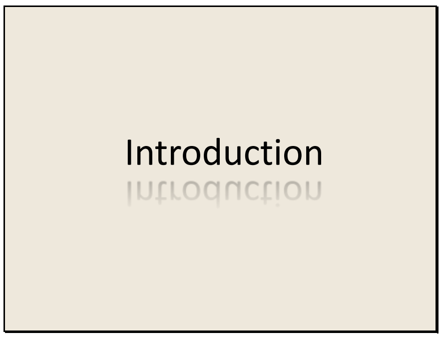
Slide 1

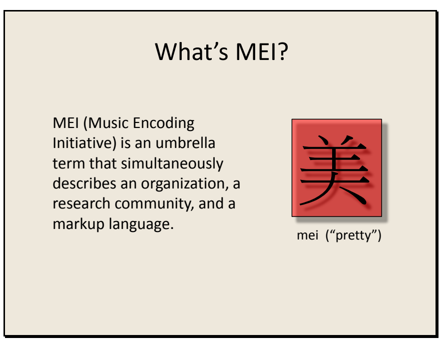
Slide 2



At the 2007 "TEI at 20" meeting in College Park, Maryland, MEI was presented in the "Outside the Box" session as a new direction for TEI. Since that time, significant events have taken place in the development of TEI with regard to music – the addition of elements for music notation, formation of a music special interest group, and the release of a TEI customization that embeds MEI.

Since MEI has reached its own 15th birthday, this seems like a good time to reflect on MEI's history and consider future possibilities. In just a few short years, MEI has moved from an "outside the box" dream toward becoming a *de facto* standard for academic, research-oriented music encoding projects. This presentation will provide a brief introduction to MEI, some reflections on its history, and a look into its future.

Slide 3



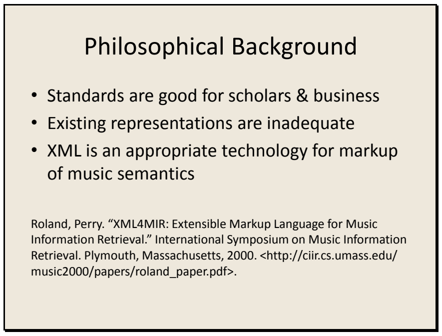
A little background –

The markup language aims to capture music notation documents in a standard, non-propriety format.

The acronym is pronounce-able – "mei" (may) is the Pinyin transliteration of the Mandarin word for "pretty". It's also a common phoneme in German, as in "meise" (titmouse).

MEI brings together specialists from various music research communities, including technologists, librarians, historians, and theorists in a common effort to define best practices for representing a broad range of musical documents and structures. The results of these discussions are formalized in the MEI schema, a core set of rules for recording physical and intellectual characteristics of music notation documents expressed as an eXtensible Markup Language (XML) schema. It is complemented by the MEI Guidelines, which provide detailed explanations of the components of the MEI model and best practices suggestions.

Slide 4



Standards extend the scale, breadth, and accessibility of scholarly evidence and encourage innovation in learning and teaching. In addition, standards facilitate innovation and collaboration in scholarly discourse.

Most existing music representations are inappropriate due to their scope. The analytical domain, which we seek to exploit in MIR, is most often the first thing to be defined as “out of scope”. Many representations define their approach to music encoding too narrowly, concentrating on a particular use of the data such as printing or automated performance. Other representations, such as the Standardized Music Description Language, have attempted to represent music too broadly.

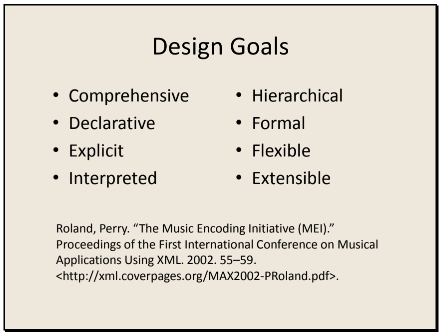
Many existing solutions are hardware or software dependent. Many existing codes are also proprietary. Therefore, their use for information exchange is severely limited.

XML is a platform-independent, open standard. Despite its name, XML isn't really a markup language, but a meta-language, designed to support the definition of community-specific languages. Because there are no limits on the use of elements across multiple namespaces or on the structural depth that a markup language might employ, XML is very powerful. Furthermore, it is easy to implement. Developers need not create their own tools for authoring, parsing, transforming or displaying XML. There is already an ever-growing set of free tools available.

Outside the music field there is a large, organized XML community already in existence. XML documents can be validated using a Document Type Definition or DTD, which is a formal statement of the rules governing the document’s grammar. Utilizing a grammar speeds encoding and reduces encoding errors. Grammars also provide a generative model. XML is a declarative representation making it preferable to a procedural one; that is, it can easily be examined and combined with information from other sources. A declarative representation is also composable, e.g. the meaning of a complex expression is based on or can be derived from the meaning of its parts and their combinations. Also, since no interactions occur between structural entities, the representation is extremely modular. The tree data structure is conceptually easy and provides efficient, non-linear data retrieval. This structure also makes it possible to apply transformations to groups of objects.

XML is extensible, an absolute requirement for music representation systems. Extensibility functions to “resolve ambiguity already latent within the existing scheme” (Huron1992, 35). XML is also human-readable. Human-readability makes data creation and maintenance easier and functions as a protection against technological obsolescence.

Slide 5



Comprehensive

Simple codes which represent only the information necessary for a particular application seem to be more efficient because they require less development effort and less complex processing software; however, a comprehensive code is better able to capture the interdependency of the elements within a score. Also, because the simple encoding method requires a new encoding for each application, a comprehensive coding scheme is "more conducive to the establishment of a permanent data base of encoded musical scores, and to the ultimate prevention of duplicated effort”.

Declarative

Declarative representations, i.e. those which state knowledge *about something, are preferred over procedural* ones, i.e. those which state *how to do something.* Declarative knowledge can be examined and combined while procedural knowledge tends to be inaccessible. Declarative representations are more modular because they limit interactions between separate entities. In addition, knowledge can be added to a declarative representation easily while procedural representations allow addition only by modification followed by a required debugging process.

Explicit

In so far as possible, all relations and knowledge should be explicitly stated in the representation. When a representation is declarative and explicit, it is naturally static. This is not to say that the representation cannot change. Indeed, the encoding may be the end result of a process, but the encoding does not represent the process, only the result.

Interpreted

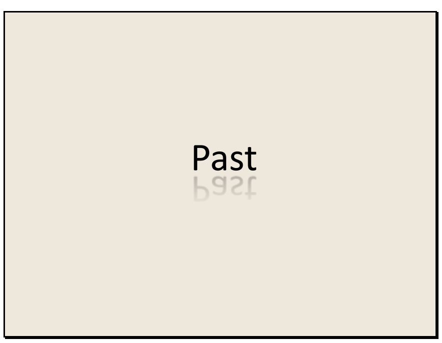
In order to encode something, one must first determine what is to be represented by the encoding. The resulting encoding is an interpretation of the thing to be encoded. In other words, whenever an attempt to assign meaning is made, interpretation occurs. A well-designed representation acknowledges this truism and allows one to make interpretations explicit.

Hierarchical

Many representation schemes have tended toward one of two extremes: encoding scores as a collection of notes or as a single entity without further division. However, there are many musically important structures that fall between these two extremes. The existence of multiple levels of structure implies the need for hierarchical representation. There are several benefits in representing musical notation hierarchically. In a hierarchical representation, individual components are isolated, making it possible to limit interactions between components and to specify the scope of operators that act on them. In addition, this kind of structure allows any component of a score to be treated in exactly the same manner as any other, regardless of its size or position in the hierarchy. In other words, a hierarchical data structure is object-oriented. However, unnecessary complexity is not introduced by using a hierarchical structure instead of the list structure commonly used to represent music. A list structure can always be represented hierarchically if desired, i.e. by a tree having only 1 level.

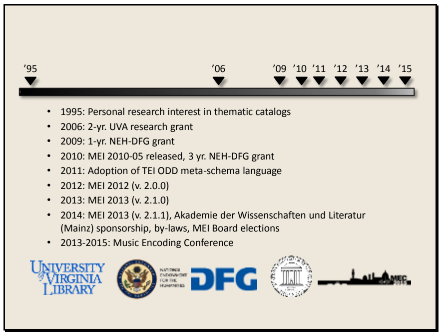
Formal
A music representation should be as formal as possible. That is, the ability to prove the correctness of the encoding should not depend on knowledge outside the formal definition. While it cannot model semantics, at least a DTD formally declares the syntax of the representation. A representation based on a formal grammar like that embodied in a DTD can describe a broad range of music. In fact, any music that can be segmented can be described by a grammar. Grammars are used extensively in a variety of disciplines. Therefore, a great deal of software, including parsers and compilers, has been developed around them. Instead of attempting to accommodate archaic and incorrect practice, the MEI should continue efforts to modernize and standardize notational practices like those contained in Read.
Flexible
While a music representation should have a standardizing effect on the corpus for which it designed, it should not strive to eliminate all variation. It must remain flexible enough to accommodate minor variations in the source material. In addition, much of the encoding should be made optional so that the encoder is not required to mark up things with which he is not concerned. However, increased flexibility must be carefully weighed against an inevitable corresponding decrease in standardization.
Extensible
No representation can guarantee that it can be used for all future artifacts or anticipate all of its own future uses, especially if the representation is the result of a solitary effort. Therefore, extensibility is required. In addition, it may be necessary to extend the representation in order to resolve ambiguity already latent in the scheme.

Slide 6



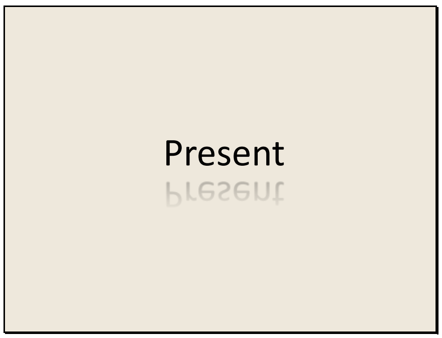
Where have we been?

Slide 7



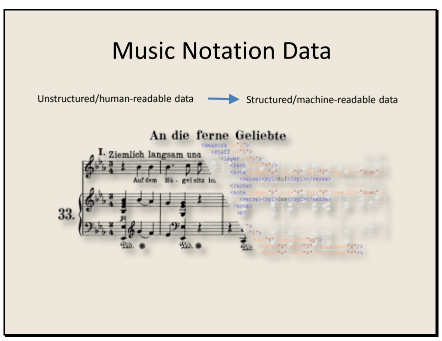
A brief history of MEI

Slide 8



Where are we now?

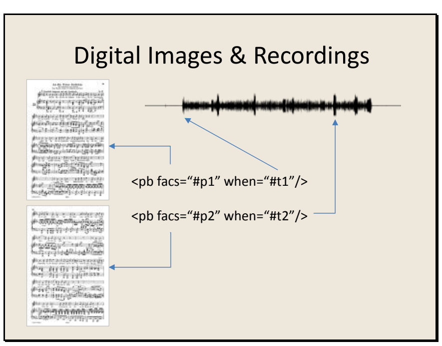
Slide 9



Encoding permits a shift away from musical notation as an unstructured digital image or audio file without machine-readable content toward various other formats, whether written, sounded or realized in other ways.

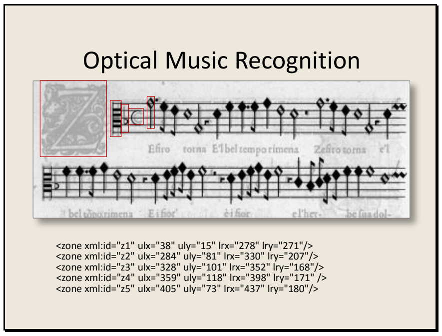
MEI goals: not limited to CMN, idiomatic XML
XML better than binary: human-readable, self-documenting, more easily preserved

Slide 10



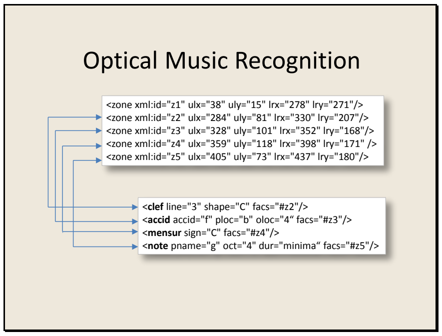
Because <pb/> references facsimile image and time point, it effectively creates an association between an image and an time point that can be traversed in either direction. Of course, this capability is not limited only to pages. It is available on nearly every entity within MEI.

Slide 11



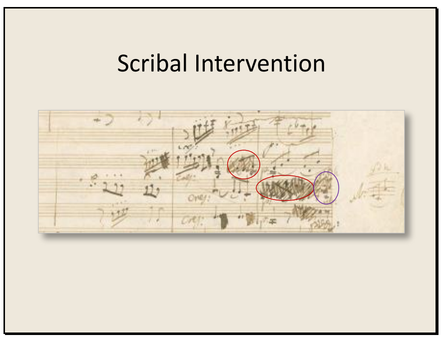
Notational features, such as notes, chords, and rests can be associated with bounding boxes containing these features within a digital image.

Slide 12



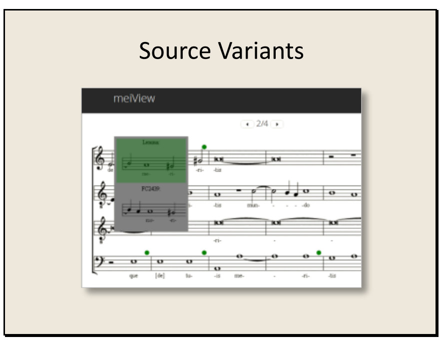
Bi-directional linkages can be created between notational features and **regions within digital images**, potentially providing information about the location of every significant feature on the page.

Slide 13



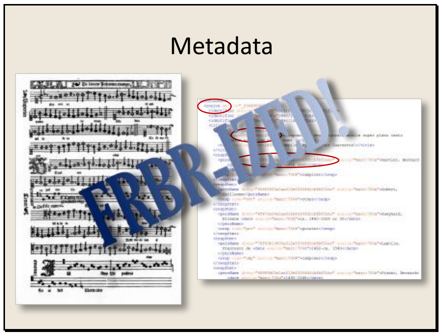
Scribal interventions, such as deletions and additions, can be captured. Red ovals indicate deletions, purple oval is an addition.

Slide 14



In meiView, green dots indicate points of divergence between the sources, in this case between the “approved/accepted” text and source FC2439. Clicking on the version presented in FC2439 inserts that version into the displayed notation, giving the user the power to create a personalized edition that fits his needs.

Slide 15



Highly detailed bibliographic and other metadata can be captured to support complex research purposes. Illustrated here: mapping between MEI and other metadata formats (MARC in this case), and authority data. While not in the form of an RDF triple, a relationship is created between the persName entity, its content, and the authorized form pointed to by the authURI and dbkey attributes.

Basic XSLT stylesheets exist for transforming MARC records into MEI and vice versa, allowing data interchange between musicological and library-oriented applications.

MEI metadata is FRBR-ized, meaning it can capture work, expression, manifestation, and item level metadata and relationships.

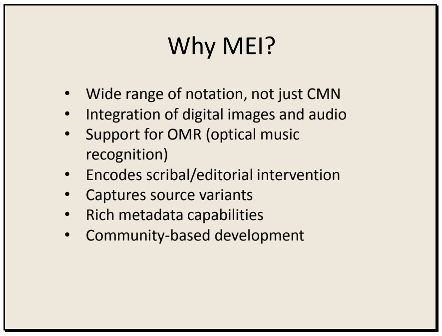
Slide 16

Community-based development is a defining feature of MEI.

Slide 17

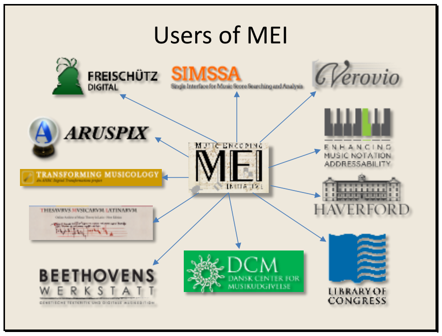
“Free” beer, as anyone who’s helped a friend move knows, often comes with conditions. A case can be made that software that doesn’t provide these rights isn’t really “free”.

Slide 18



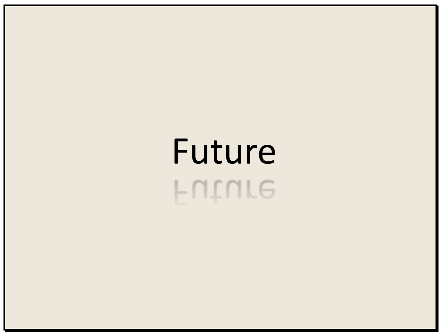
A better question may be, Why not MEI?!

Slide 19



MEI users are primarily researchers dissatisfied with the status quo; that is, proprietary representations and tools without the scope and power of MEI that focus on common practice notation and the paper publication paradigm.

Slide 20

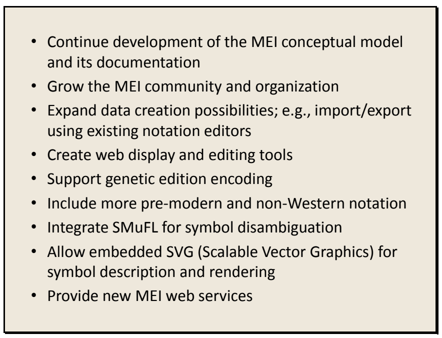


Where are we going?

Slide 21

Obviously, there have been challenges along the way. But, rather than talk about past “challenges”, I believe it’s more fruitful to speak of future “opportunities”.

Slide 22



Some of the opportunities MEI will address in the future:

- Continue development of the MEI conceptual model
    - Next release: 36 changes, 6 of which break backward compatibility
    - However, number & scope of changes decreasing
- Grow MEI community
    - Continue relationship with Akademie der Wissenschaften und der Literatur in Mainz, an MEI sponsor
    - Provides opportunities for interfacing w/ other Akademie-related endeavors, such as RISM and critical edition projects
    - Continue development of by-laws, elect new Board members, create additional interest groups
    - Move repository from Google Code to GitHub, perhaps leading to increased participation in MEI development
- Expand data creation possibilities
    - Continue work on SibMEI (MEI export plug-in for Sibelius) and similar export methods for other notation software
    - Work on existing and new translation programs, such as those from MusicXML, Humdrum **kern, SCORE, etc.
- Create web display and editing
    - Use Verovio as the basis for an web-based MEI editor
- Support genetic edition encoding
    - Will allow tracing the “writing acts” in manuscripts
- Include pre-modern and non-Western notation
    - Expand the coverage of MEI to earlier Western as well as non-Western notation repertoires, such as black mensural notation
    - Consider how these repertoires are similar to/different from 20th and 21st c. developments in notation
- Integrate SMuFL (Standard Music Font Layout)
    - Addresses symbol disambiguation (semantic domain) and alleviates some display issues (visual

domain)

- Allow embedded SVG (Scalable Vector Graphics)
    - Permits capturing precise rendition of non-standard symbols, non-notation writing acts (i.e., editorial signs, such as strikeouts and meta-marks)
    - Provides mechanism for embedding font descriptions inside MEI (for data sharing and preservation purposes)
- Provide additional MEI-based web services
    - More fine-grained control over schema customization
    - Content-based indexing/search applications

Slide 23

If these topics (and others) interest you, please join us for the 4th iteration of the Music Encoding Conference. Papers and posters are welcome on all topics under the umbrella of "music encoding", not only MEI-related submissions.

Slide 24